

# ARCHI2 - Compte-rendu du TME1

Nicolas Phan

pour le 16 Février 2018

## Table des matières

<b>1</b>	<b>Automate du composant PibusSimpleRam</b>	<b>2</b>
1.1	Question C1 . . . . .	3
1.2	Question C2 . . . . .	3
<b>2</b>	<b>Automate du composant PibusSimpleMaster</b>	<b>4</b>
2.1	Question D1 . . . . .	6
2.2	Question D2 . . . . .	7
<b>3</b>	<b>Automate du composant PibusSegBcu</b>	<b>8</b>
3.1	Question E1 . . . . .	9
3.2	Question E2 . . . . .	9
3.3	Question E3 . . . . .	9
<b>4</b>	<b>Modélisation de l'architecture matérielle</b>	<b>9</b>
4.1	Question F1 . . . . .	9
4.2	Question F2 . . . . .	10
<b>5</b>	<b>Simulation</b>	<b>10</b>
5.1	Question G1 . . . . .	10
5.2	Question G2 . . . . .	10
5.3	Question G3 . . . . .	10
5.4	Question G4 . . . . .	10
5.5	Question G5 . . . . .	11

# 1 Automate du composant PibusSimpleRam

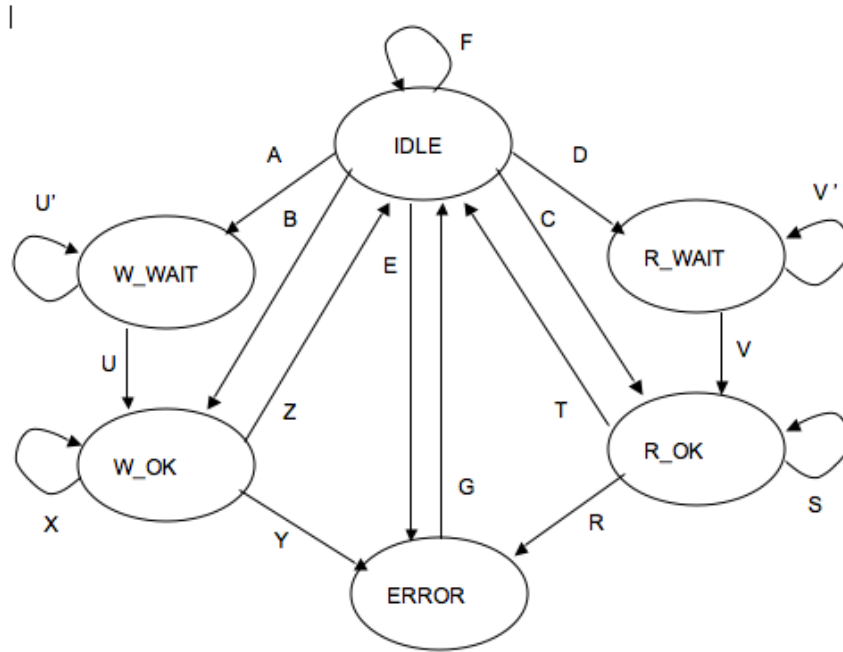


FIGURE 1 – Graphe de la MAE du composant RAM

Etat	Signification
IDLE	La RAM est inactive, elle n'est la cible d'aucune transaction
W_WAIT	Une transaction d'écriture dans le composant RAM est en cours, l'automate attend que la RAM soit disponible.
W_OK	Une transaction d'écriture dans le composant RAM est en cours, la RAM effectue l'écriture.
R_WAIT	Une transaction de lecture dans le composant RAM est en cours, l'automate attend que la RAM doit disponible
R_OK	Une transaction de lecture dans le composant RAM est en cours, la RAM a effectue la lecture
ERROR	Une erreur s'est produite

TABLE 1 – Description des états de la MAE

## 1.1 Question C1

Nom	Transition	Explication
A	$\overline{\text{SEL}} . \text{ADR\_OK} . \text{READ} . \text{DELAY}$	Une requête d'écriture a été reçue, la RAM est latente donc l'automate attend.
B	$\overline{\text{SEL}} . \text{ADR\_OK} . \text{READ} . \overline{\text{DELAY}}$	Une requête d'écriture a été reçue, l'écriture s'effectue.
C	$\overline{\text{SEL}} . \text{ADR\_OK} . \text{READ} . \overline{\text{DELAY}}$	idem en lecture
D	$\overline{\text{SEL}} . \text{ADR\_OK} . \text{READ} . \text{DELAY}$	idem en lecture
E	$\overline{\text{SEL}} . \text{ADR\_OK}$	Le composant RAM est la cible d'une transaction comportant une adresse illégale.
F	$\overline{\text{SEL}}$	Le composant RAM n'est pas sélectionné pour une transaction, il reste inactif.
G	1	Il n'y a qu'une seule transition sortant de ERROR, elle est donc inconditionnelle.
U	$\overline{\text{G0}}$	La RAM est latente, l'automate reste en état d'attente.
U'	G0	La RAM effectue l'écriture, le composant est à l'état READY
V	$\overline{\text{G0}}$	idem en lecture
V'	G0	idem en lecture
X	$\overline{\text{SEL}} . \text{ADR\_OK} . \text{READ}$	Le composant RAM est toujours sélectionné, ses entrées n'ont pas changé
Y	$\overline{\text{SEL}} . (\text{ADR\_OK} + \text{READ})$	Le composant RAM est toujours sélectionné mais ses entrées ont été corrompues
Z	$\overline{\text{SEL}}$	La transaction se termine, le composant RAM n'est plus sélectionnée
R	$\overline{\text{SEL}} . (\text{ADR\_OK} + \text{READ})$	idem en lecture
S	$\overline{\text{SEL}} . \text{ADR\_OK} . \text{READ}$	idem en lecture
T	$\overline{\text{SEL}}$	idem en lecture

TABLE 2 – Fonctions de transition de la MAE de SimpleRam

## 1.2 Question C2

	ACK_EN	ACK_VALUE	DT_EN	MEM_CMD
IDLE	0	WAIT	0	NOPE
R_WAIT	1	WAIT	0	READ
R_OK	1	READY	0	READ
W_WAIT	1	WAIT	1	WRITE
W_OK	1	READY	1	WRITE
ERROR	1	ERROR	0	NOPE

TABLE 3 – Valeurs des signaux de sortie de la MAE de SimpleRam

## 2 Automate du composant PibusSimpleMaster

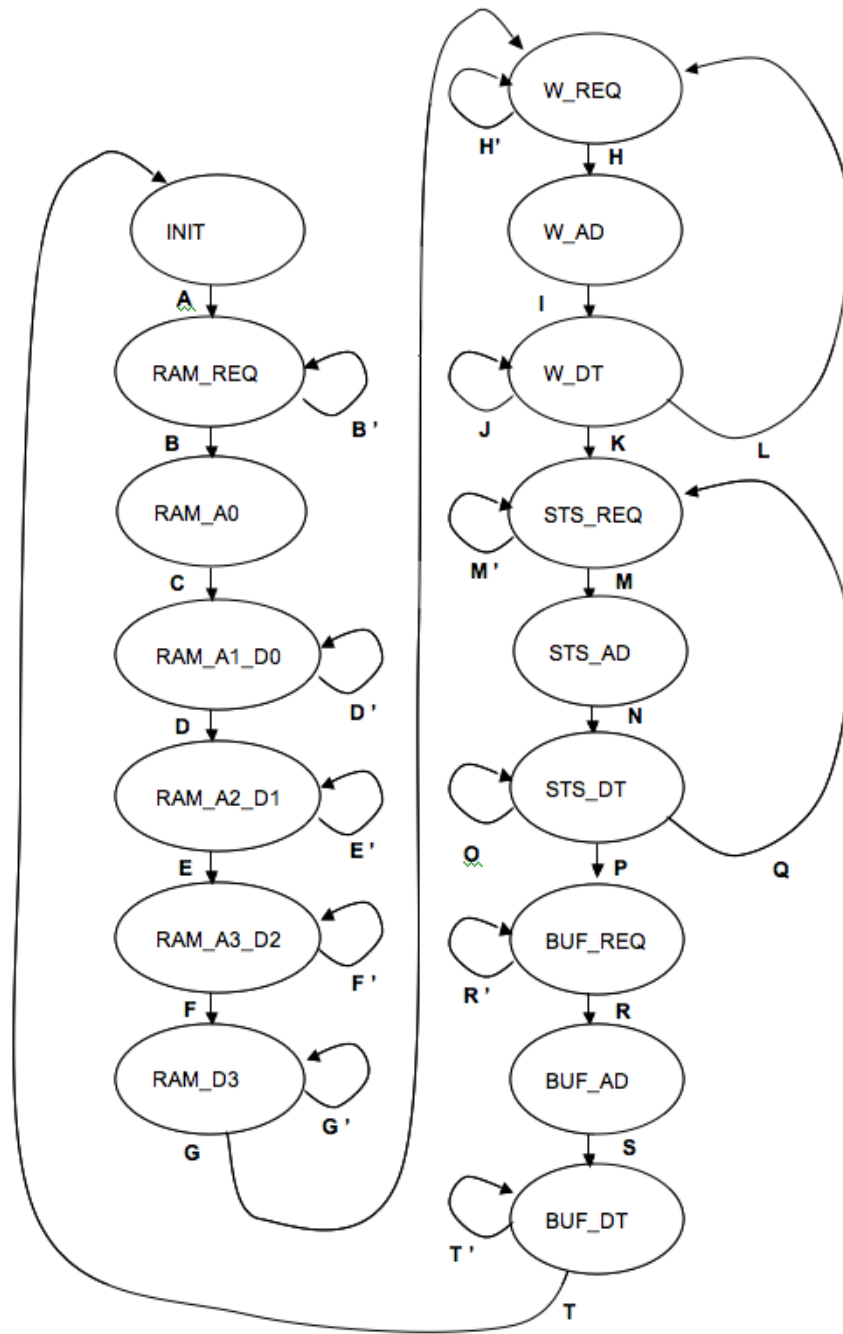


FIGURE 2 – Graphe de la MAE du composant Master

Etat	Signification
INIT	Etat initial
RAMREQ	Le maitre demande l'accès au bus pour y effectuer une transaction rafale de lecture de 4 mots dans la RAM.
RAMA0	Le maitre vient d'obtenir le bus, il fournit la première adresse
RAMA1D0	Le maitre envoie une adresse qui n'est pas la première de la transaction : il effectue un transfert rafale.
RAMA2D1	Idem, ici le maitre envoie la troisième adresse de la rafale
RAMA3D2	Idem, ici le maitre envoie la quatrième adresse de la rafale
RAMD3	Le maitre attend la dernière réponse de la RAM.
WREQ	Le maitre demande l'accès au bus pour y effectuer une transaction d'écriture dans le registre DISPLAY du composant TTY dans le but d'afficher un caractère à l'écran.
WAD	Le maitre vient d'obtenir le bus il fournit sa commande.
WDT	Le maitre attend la réponse de la cible
STSREQ	Le maitre demande l'accès au bus pour une transaction de lecture du registre STATUS du périphérique TTY afin de déterminer si un caractère a été entré au clavier.
STSAD	Le maitre vient d'obtenir le bus il fournit sa commande.
STSDT	Le maitre attend la réponse de la cible
BUFREQ	Le maitre demande l'accès au bus pour une transaction de lecture du registre KEYBUF du TTY afin de connaître le caractère entré au clavier par l'utilisateur.
BUFAD	Le maitre vient d'obtenir le bus il fournit sa commande.
BUFDT	Le maitre attend la réponse de la cible

TABLE 4 – Description des etats de l'automate

## 2.1 Question D1

Nom	Transition	Description
A	1	Une seule transition sort de l'état INIT, elle est donc inconditionnelle
B'	$\overline{\text{GNT}}$	Le composant master demande le BUS tant qu'il ne l'a pas obtenu.
B	GNT	Le maitre a obtenu le bus, il va envoyer sa première commande
C	1	D'après le protocole PIBUS, lors d'une transaction rafale, la première commande est toujours enregistrée par la cible, donc le maitre passe inconditionnellement à la deuxième commande une fois que la première est envoyée.
D'	RDY	Tant que la cible n'a pas acquitté, le maitre continue d'envoyer sa commande.
D	$\overline{\text{RDY}}$	Le maitre a eu l'acquiescement de la cible, il passe à la commande suivante.
E	RDY	idem
E'	$\overline{\text{RDY}}$	idem
F	RDY	Le maitre a bien reçu l'avant-dernière réponse, il ne lui reste plus qu'à attendre la dernière réponse.
F'	$\overline{\text{RDY}}$	idem
G	RDY	Le maitre a reçu la dernière réponse, la transaction se termine, il passe à la transaction suivante.
G'	$\overline{\text{RDY}}$	Le maitre reste en état d'attente tant qu'il n'a pas reçu la dernière réponse.
H'	$\overline{\text{GNT}}$	même situation que B'
H	GNT	même situation que B
I	1	Le maitre a envoyé sa commande, il va attendre la réponse
J	$\overline{\text{RDY}}$	Le maitre attend sa réponse de la cible tant qu'il ne l'a pas eue.
K	RDY.LAST	Le maitre a obtenu une réponse de la cible (RDY, on ne traite pas les cas d'erreur) et le caractère envoyé était le dernier, le maitre passe à l'étape suivante.
L	RDY. $\overline{\text{LAST}}$	Idem, sauf qu'il ne s'agit pas du dernier caractère, le maitre engage une nouvelle transaction pour envoyer le caractère suivant.
M'	$\overline{\text{GNT}}$	même situation que H'
M	GNT	même situation que H
N	1	même situation que I
O	$\overline{\text{RDY}}$	même situation que J
P	RDY.NUL	Le maitre a obtenu le contenu du registre STATUS, et ce-dernier indique qu'un caractère a été entré.
Q	RDY.NUL	Aucun caractère n'a été entré, le maitre continue la lecture en boucle de STATUS.
R'	$\overline{\text{GNT}}$	même chose que H'
R	GNT	même chose que H
S	1	même chose que I
T'	$\overline{\text{RDY}}$	La transaction est terminée
T	RDY	même chose que J

TABLE 5 – Fonctions de transitions de la MAE de SimpleMaster

## 2.2 Question D2

	REQ	CMD_EN	ADR_VALUE	READ_VALUE	LOCK_VAL	DT_EN
INIT	0	0	X	X	X	0
RAM_REQ	1	0	X	X	X	0
RAM_A0	0	1	ram_base	1	1	0
RAM_A1D0	0	1	ram_base + 4	1	1	0
RAM_A2D1	0	1	ram_base + 8	1	1	0
RAM_A3D2	0	1	ram_base + 12	1	1	0
RAM_D3	0	0	X	X	0	0
W_REQ	1	0	X	X	X	0
W_AD	0	1	seg_tty_base	0	0	0
W_DT	0	0	seg_tty_base	0	0	1
STS_REQ	1	0	X	X	X	0
STS_AD	0	1	seg_tty_base + 4	1	0	0
STS_DT	0	0	seg_tty_base + 4	1	0	0
BUF_REQ	1	0	X	X	X	0
BUF_AD	0	1	seg_tty_base + 8	1	0	0
BUF_DT	0	0	X	X	X	0

TABLE 6 – Valeurs de sortie de la MAE de SimpleMaster

### 3 Automate du composant PibusSegBcu

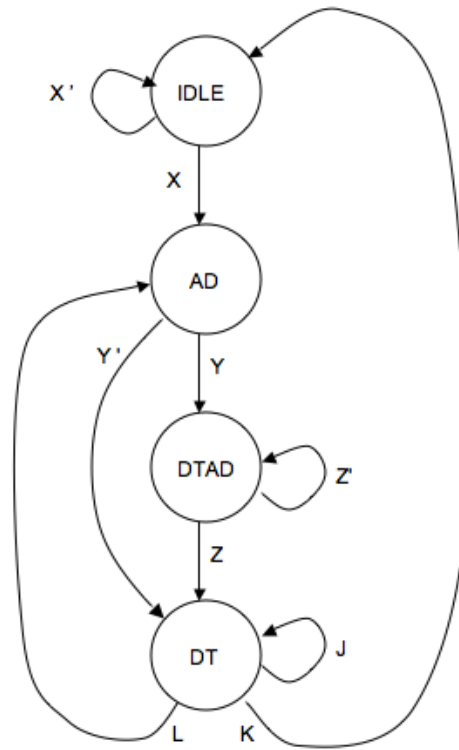


FIGURE 3 – Graphe de la MAE du composant BCU

Etat	Description
IDLE	Le BCU est libre et attend une requête, le BCU et le maitre sont inactifs
AD	Le bus vient d'être alloué au maitre, la première commande est envoyée
DTAD	Le bus est alloué à un maitre, il envoie une commande qui n'est pas la première de la transaction en cours (le maitre est en train d'effectuer une transaction rafale)
DT	Le maitre attend la dernière réponse de la cible, le bus ne lui est plus alloué

TABLE 7 – Description des états de l'automate



### 3.1 Question E1

Nom	Transition	
X	$\overline{\text{REQ}}$	Tant que l'unique maitre ne demande pas le bus, le BCU reste inactif.
X	REQ	Le maitre a demandé le BCU, comme il est le seul maitre ici, il obtient le bus.
Y	LOCK	Il s'agit d'une transaction rafale
Y'	$\overline{\text{LOCK}}$	IL ne s'agit pas d'une transaction rafale, l'automate va simplement attendre la réponse de la cible.
Z'	$\text{LOCK} + (\text{ACK} \neq \text{WAIT})$	Tant que le maitre n'a pas envoyé toutes ses commandes ou n'a pas reçu d'acquittement à celles-ci (excepté pour la dernière), la rafale continue.
Z	$\overline{\text{LOCK}} \cdot (\text{ACK} \neq \text{WAIT})$	Le maitre a envoyé sa dernière commande et a reçu l'acquittement pour les commandes précédentes, il passe à l'état d'attente de la dernière réponse.
J	$(\text{ACK} \neq \text{WAIT})$	Le maitre continue d'attendre la dernière réponse tant qu'il ne l'a pas eue.
K	$(\text{ACK} \neq \text{WAIT}) \cdot \overline{\text{REQ}}$	La transaction est terminée, et le maitre n'en demande pas d'autre
L	$(\text{ACK} \neq \text{WAIT}) \cdot \text{REQ}$	La transaction est terminée, et le maitre redemande une transaction, le BCU lui redonne immédiatement le BUS.

TABLE 8 – Fonctions de transition de SegBcu

### 3.2 Question E2

	GNT	SEL0	SEL1
IDLE	REQ	0	0
AD	1	$A \in \text{zoneRAM}$	$A \in \text{zoneTTY}$
DTAD	0	$A \in \text{zoneRAM}$	$A \in \text{zoneTTY}$
DT	$\overline{\text{WAIT}} \cdot \text{REQ}$	$A \in \text{zoneRAM}$	$A \in \text{zoneTTY}$

TABLE 9 – Valeurs de sortie de SegBcu

### 3.3 Question E3

Pour que les bus qui respectent le protocole PIBUS effectuent les transactions le plus rapidement possible, le protocole spécifie que dans certains cas, une phase d'une transaction peut commencer avant qu'une autre phase de la transaction précédente soit terminée : le bus est pipeliné. Ici, le bus peut entamer une phase d'allocation pendant la phase de réponse de la transaction précédente, c'est pour cela qu l'allocation est réalisée aussi dans DT.

## 4 Modélisation de l'architecture matérielle

### 4.1 Question F1

A la construction du composant master sont fournies les adresses de début des segments de l'espace adressable correspondant à la RAM et aux registres TTY, respectivement définis par les macros SEG\_RAM\_BASE et SEG\_RAM\_TTY.

Le constructeur de RAM prend 0 en tant que target id car ici, la RAM est la cible 0 (et le TTY la cible 1), il prend une référence vers l'objet segtable, une variable définissant le nombre de cycle de latence de la RAM puis un objet de type Loader permettant de charger des données en RAM.

```

1  PibusSegBcu      bcu      ("bcu", segtable, 1, 2, 100);
2  PibusSimpleMaster master ("master", SEG_RAM_BASE, SEG_TTY_BASE);
3  PibusSimpleRam   ram      ("ram", 0, segtable, ram_latency, loader);
4  PibusMultiTty    tty      ("tty", 1, segtable, 1);

```

Ici il s'agit d'affecter les bons signaux aux entrées/sortes des composants.

```

1  master.p_ck      (signal_ck);
2  master.p_resetn  (signal_resetn);
3  master.p_gnt     (signal_gnt_master);
4  master.p_req     (signal_req_master);
5  master.p_a       (signal_pi_a);
6  master.p_opc     (signal_pi_opc);
7  master.p_read    (signal_pi_read);
8  master.p_lock    (signal_pi_lock);
9  master.p_d       (signal_pi_d);
10 master.p_ack     (signal_pi_ack);
11 master.p_tout    (signal_pi_tout);
12
13 ram.p_ck         (signal_ck);
14 ram.p_resetn     (signal_resetn);
15 ram.p_sel        (signal_sel_ram);
16 ram.p_a          (signal_pi_a);
17 ram.p_read       (signal_pi_read);
18 ram.p_opc        (signal_pi_opc);
19 ram.p_ack        (signal_pi_ack);
20 ram.p_d          (signal_pi_d);
21 ram.p_tout       (signal_pi_tout);

```

## 4.2 Question F2

Le segment de l'espace adressable correspondant aux registres du composant TTY début à l'adresse définie dans `SEG_TTY_BASE` et est de taille 16 octets. En effet, il y a 4 registres adressables dans le TTY, et ces registres font 4 octets, il y a donc 16 octets adressables, d'où le 0x00000010.

```

1  segtable.addSegment("seg_tty", SEG_TTY_BASE, 0x00000010, 1, false);

```

# 5 Simulation

## 5.1 Question G1

La simulation de 10 millions de cycles nécessite 13 secondes.

## 5.2 Question G2

Le maitre reste en état de demande du Bus pendant un seul cycle, il obtient le bus au cycle suivant, il n'y a donc pas d'état d'attente. Ceci est dû au fait qu'il n'y a pas d'autres maitres sur le bus, étant donné qu'il est le seul maitre susceptible de demander l'accès au bus, le BCU n'a pas de raison de le lui refuser.

## 5.3 Question G3

Le maitre attend la réponse de la RAM pendant deux cycles, pendant lesquels la RAM est à un état WAIT, ceci est dû à la latence de la RAM qui a été fixée à 2 cycles dans le prototype virtuel de l'ensemble.

## 5.4 Question G4

L'affichage d'un caractère sur le TTY prend 3 cycles qui sont les cycles d'allocation, commande et réponse de la transaction simple d'écriture effectuée par le maitre pour écrire dans le registre `DISPLAY` du TTY.

## 5.5 Question G5

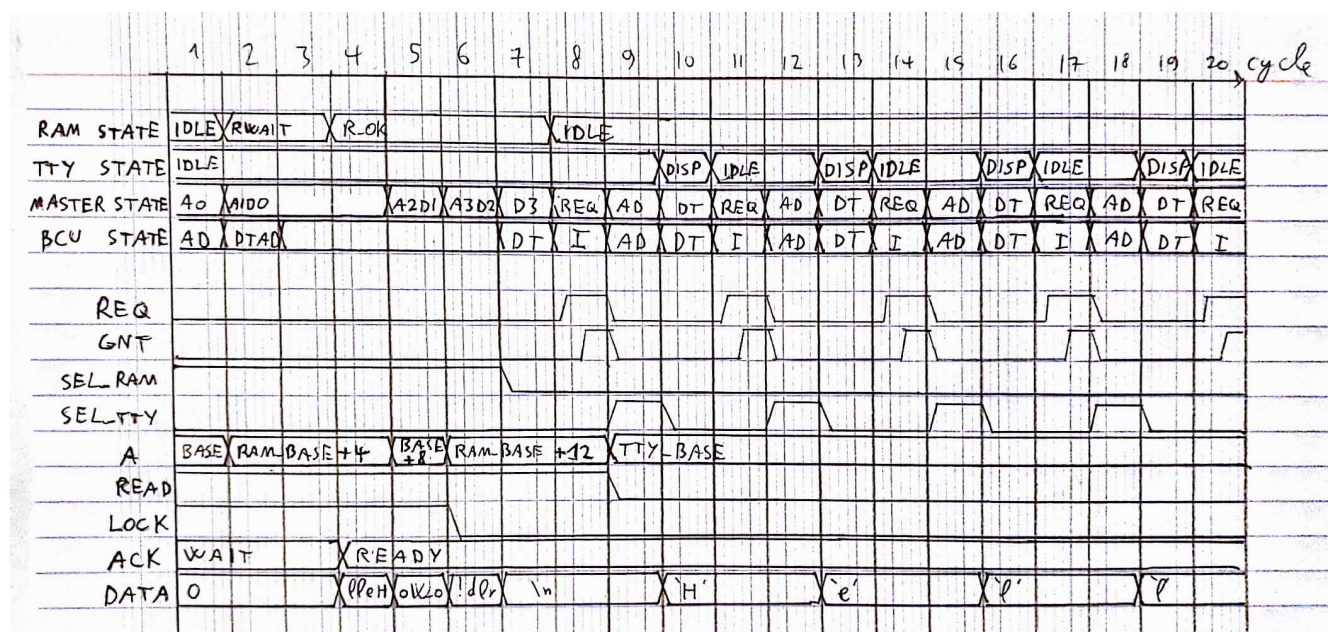


FIGURE 4 – Chronogramme des signaux du PIBUS